

invisible debugger 8 dec 1969

nbp=4
dfd=65

```
dimension sbu(40)      /small buffer
dimension syl(2)        /octal number, decimal number
dimension let(1)         /becomes 0 if letter seen
dimension chi(1)         /becomes 0 if any syllable
dimension chc(1)         /character packing count
dimension ch(1)          /character
dimension sym(1)         /pointer to symbol being packed
dimension wrd(1)          /expression
dimension cnc(1)          /concise code of symbol
dimension df1(1)          /value for symbol definition
dimension l1f(1)          /nonzero if A I etc. typed
dimension fa(1)           /expression preceding last <
dimension fa2(1)          /expression preceding next to last <
dimension mod(1)          /mode, -0 for type-in
dimension tas(1)          /current examine address
dimension pvp(1)          /field, set up by trc
dimension dff(1)          /switch used in fet, dep
dimension t0(1),t1(1),t2(1),t3(1),t4(1)
```

100/
100, jmp nt0 /entry from user
101, jmp ere /ill inst in ID
102, jmp lse-2 /hit call in ID
lse-2, clc
dac whr
lse, jsp lcc
lse+1, eem
iam
law squoze i
dac low /repair symbol table
lsa, dzm mod
dzm tas
lss, dzm iif
cla
lss+2, dap xsw /clear `<` switch
law lwt
dap bax
clc \downarrow clf 7
dac chi
dzm wrd
lac (jmp lse
dac aus
lac (010000
dac pvf /initialize pv
law top-bbu
add bot
and (7740
dap pg8 /max count for pv
lac lwt
dac df1
lac (ior i
ssn, dip sgn
dzm syl
dzm syl+1
n2, law i 2
add bot
dac sym
TAX
dzm i top
dzm cnc
clc \downarrow stf 1 /enable call button
dac let
dac chc

lsr, tyi
lsr+1, dio ch
TIX
lac i dtb
cas, skp /skip if lower case
rar 9s
and (777
dac t2
dac t4
sub (44
spa
jmp ln
add (jmp uc
dap lsz

```
sub ari      /last no-eval routine
lio chi
spa
jmp i lsx
lxr sym
lac i tpp
lio lcp
sad (squoze ,
dio iif    /syllable was .
lxr (syl-top
lio let
spi i
jdp evl
jmp ev4
law 77
sad ch
dzm tas
lac (flexo U
jda tys
jmp lss

ev4,      lac lsx
sub cbr
spa
jmp . 4
lac iif
sza
jmp err
lac wrd
sgn,      top      /operator
dac wrd
dac df1
lio chi
spi
lac lwt
dac t0
jmp .

lsx,
```

n, rir 5s /digit routine
spi
dzm t4
lxr (-2
lac ops
mul i syl+2
scr 1s
A+II
lac t4
A+II
sza↓sni
cli↓cmi /-0
dio i syl+2
law 10.
SXXP
jmp .-12
jmp l1
ln, add (44-12
spa
jmp n

l1, dzm let /letter routine
dzm chi
idx chc
sas (4
jmp ln3
law 1
dac chc
law i 1 /move rest of symbol down
adm sym
TAAx
add (top-bbu
spa
jmp ser-3 /no room
lac i top+1
dac i top
SXXA
sas bot
jnp .-4
dzm i top-2
lac (400000
adm i top-3
dzm cnc
ln3, lxr bot
lac i top-2
mul (50
div (1
hlt
add t2
SAA
dac i top-2
lac cnc
ral 6s
ior ch
dac cnc
jmp lsr

mpi, law i 7777 /read mpr instruction.
dac syl
dzm chi
upw-2, law up1-1
dap upq
upw, tyi
law charac r/
A\$IP|
jmp del
upw+4, idx upq
TAX
law i 7777
and i 0
sad (jmp
upq, jmp .
ral 6s
A\$IP
jmp upw+4
law 7740
and i 0
scr 6s
spi i
ral 6s
xor syl
dac syl
law 37
and i 0
add upw-2
dap upq
jmp upw

/bug - if lower case typed, goes to err, cas is wrong

```
define pack a)b,c,d
char l'a b*100 c-up1 d
termin

up1,      pack t,40,up3
          pack c,42,up3
          pack s,10,u2p
          pack z,10,up8
          pack a,60,up6
          pack i,40,up7
          pack x,20,up6
u2p,      pack a,2,up8
          pack i,4,up8
          pack x,6,up8
up3,      pack i,60,up8
          pack a,0,up8
up4,      pack x,20,up8
up5,      pack a,60,up8
          pack i,40,up8
          jmp err

up6,      pack 0,4,up5
          pack m,6,up5
up7,      pack -,16,up4
          pack 3,14,up4
          pack 6,12,up4
          pack 5,10,up4
          jmp err

up8,      pack a,1,up8
          pack i,40,up8,40
          pack x,20,up8,40
          pack ;,17,up8,40
          pack p,2,up8,40
          pack m,1,up8,40
          pack [,],3,up8,40
          pack 8,10,up8,40
          pack 7,4,up8,40
          pack :,3,up8,40
          jmp lsr+1
```

ev1, 0 /symbol lookup
lio bot
jmp .+3
ev1+3, TXIP|
jmp evn /not found
lxr sym
ev2, lac i top
X->IX
SXX
sad i top-1
jmp ev3
sma
jmp ev3-3
ev5, lac i top-1
SXX
spa
jmp ev5
jmp ev1+3
ev3-3, ior (200000 /check for calm symbol
sas i top-1
jmp ev5
ev3, SII
sma
jmp i ev1
X->IX
jmp ev2
evn, idx ev1
jmp i ev1

/no-eval routines

uc,	ZAP	/upper case
lc,	law 600	/lower case
	dap cas	
	jmp lsr	
sqo,	spi	/
	jmp mpi	
	lxr sym	
	lac i top	
	jmp n1	
quo,	lac cnc	/^
	jmp n1	
q,	lac lwt	/Q
	jmp n1	
f,	spi i	/F
	jmp .ff	
	law fg	
n0,	lio iif	
	sni i	
	jmp err	
	dac iif	
n1,	dzm chi	
	dac syl	
	jmp n2	

daq, lac df1 /\
 jmp def-2

com, lac lcp /,
 sza
 jmp erp
 lac loc
def-2, and (7777
 dac df1

def, lac let /)
 ior iif
sk1, sza
 jmp erp
 law pn2
de, dzmdff /switch for overflow printout
 dzm syl /switch for calm mode
 dap dex
 jdp evl
 jmp df2
 law i top-bbu-300 /new symbol
 sub sym
 sma
 jmp ser /overflow
 lxr bot
 lac df1
 dac i top-1
 lac sym
 dac bot
 lac (200000
 lio syl
 sni i
 adm i top-2
 jmp dex

df2, lio df1 /redefine old symbol
 dio i top
 lac i top-1
 lio syl
 sni
 and (-200000 /leave symbol loud if it was already
 dac i top-1
 jmp .

dex, lio chc /.
 law 7777
 and loc
 spi i
 lac syl+1
 dac syl
 law squoze , -1
 dac t2
 jmp l1

del, jmp pn2 /?

/eval routines permitting A,I, etc.

pls, lac (add i /+,space
 jmp ssn

min, spi /-
 dio wrd
 lac (sub i
 jmp ssn

ovb, clc /;
 spi
 jmp n1
 lac iif
 sas (bk1
 jmp ov2
 clc /B;
 b=0
 repeat nbp,dac bk1 b b=b 1
 jmp lse

ov2, sas (msk
 jmp err
 clc /M;
 dac msk
 dzm ll
 law 7777
 dac ul
 jmp lse

bac, law opt /[
 jmp .+2

bas, law pi /]
 dap bax
 lac iif
 sza i
 jnp bar
 dac tsp
 lac t0
 jmp ta9

vb, law ta5 /|
 jmp bar+1

lpr, clc /
 dac mod

bar, law ta5+4 /slash
bar+1, dap br2
 spi i

br2, jmp .
 lac lc2
 dac ts2
 law ta6-ta5
 adm br2

cbr, jmp br2 /used as constant

/other eval routines

kill, spi i /K
 jmp ki5
 law i top-low
ki6, dac bot
 jmp lse

prc, lac lc2
eql, jdp eap /=
 jda opt
pn2, jsp lct
 jmp lss

pbx, jdp eap /\$
 jda tys
 jmp pn2

arw, jdp eap /→
 jda pi
ar1, jnp del /used as constant

oct, spi i /H
 jmp .ho
 law 10
 jmp .+4
dec, spi i /U
 jmp .un
 law 12
 dac ops
 jmp lse

smb, spi i /S
 jmp .sv
 law pi
 jmp cns+3

cns, spi i /C
 jmp clm
 law opt
cns+3, dap pns
 jmp lse

oad, spi i /O
 jmp .ob
 law poc+1
 jmp tls
rad, spi i /R
 jmp rdx
 law ped
tls, dap pa1
 jmp lse

erp, law i 47
 aqq
ere, law i 51
 arq
err, lac (743521
 jda tys
 law 7234
 jda tys
 jmp lse-2

a, law ac /A
spi
jmp n0
TAX
lac wrd
dac i 0
jmp lse

ir, law io /I
jmp a+1

sxr, law xr /X
jmp a+1

m, law msk /M
spi
jmp n0
lac wrd
xsw, skp
jmp am
dap ul
lac fa
dap ll
lac ul
sub ll
spa
jmp err
jmp lse

am, dac msk
jmp lse

bk, law bk1 /B
spi
jmp n0
add (add
dac tas
dac tsp
law 7777
and wrd
ral 6s
ior lc2
rar 6s
jda dep
jmp lse

uni, jmp ssn-1 /↓

isc, lac (and i /←
jmp ssn

ndb, spi /
jmp prc
dac lwt
and (177
sad lwt
jmp . 4
lac lwt
and (070000
ral 6s
jdp ckk
dac lc2
jmp lse

ckk, 0 /check field, AC savee
sza i
jmp i ckk /don't need to
jda trc
law 4000
lio (bbu
lxr (40
jda ree
lac trc
jmp i ckk

tab, spi i /tab
 jda dep
 dzm lcp
ta3, dac wrd
 jsp lcc
 lac wrd
 lio lcp
 sni
 jmp ta4
and (7777 /internal
sub (ac
dac spt
TAAX>P
jmp err
sub (nir
sma
jmp err
lac i aa
jda tys
lxr spt
lac i ab
jda tys
jmp ta4+1

bs, spi i /backspace
 jda dep
 law i
bs+3, add loc
 jmp ta3

fs, spi i /†
 jda dep
 law i 1
 jmp bs+3

/dispatch to routines in sch section

```
.sv:,      jsp gts
.un:,      jsp gts
.ho:,      jtp gts
.ob:,      jsp gts
.wd:,      jsp gts
.nw:,      jsp gts
.ea:,      jsp gts
.zr:,      jsp gts
.vf:,      jsp gts

gts,       dap lsx
           law sch
           sad drm
           jmp sch
           dac drm
           mta
           lio (dfd*i+bbu-sch
           law sch
           mta 104      /read drum
           jmp dre
           jmp sch
```

/dispatch to routines in run section

```
.ed:,      jsp gtr
.ff:,      jsp gtr
.bg:,      jsp gtr
.xe:,      jsp gtr
.pr:,      jsp gtr
.pw:,      jsp gtr

gtr,       dap lsx
           law sch+szh
           jmp gts+2
```

/dispatch to routines in pdv section

```
.tp:,      jsp gtp
.tb:,      jsp gtp
.jb:,      jsp gtp
.pn:,      jsp gtp
.rd:,      jsp gtp

gtp,       dap lsx
           law sch+szh+szn
           jmp gts+2
```

xec,	law xrg /* spi jmp n0 jmp .xe	
uc8,	spi i /> jda dep lio lc2 dio ts2 jmp ta6+5	
cr,	spi i /c.r. jda dep dac lwt law 72 jda tys jmp lse+1	
eee,	spi /E jmp .ed jmp .ea	
www,	law diw /W spi jmp n0 jmp .wd	
bgn,	law pc /G spi jmp n0 jmp .bg	
ttl,	spi /L jmp .tp lac let sza jmp err lac i top-1 and (-200000 jmp cm2	
put,	spi /K jmp err lio fa dio fa2 lio xsw dio xs2 dac fa law 72 jda tys law 600 jmp lss+2	
tbl,	law tys /T spi jmp cns+3 jmp .tb	

ki5, lac let /kill single symbol
sza
jmp err
TXI
dio t4
lio bot
ki1, TIIXA
sad (low-top
jmp err
lac i top
SXX
spa
jmp .-3
TXXA
sad t4
jmp ki2
SAI
jmp ki1
lac bot
A\$IP|
jmp ki3
law i 1
A+II
X→IX
lac i top
X→IX
dac i top
law i 1
A+XX
jmp ki2
SXA
jmp ki6

eap, 0 /eq1,arw,pbx common
dac lwt
jsp lct
jmp i eap

clm, lac let /make symbol calm
sza
jmp err
lac i top-1
ior (200000

cm2, dac i top-1
jmp lse

rdx, lac syl+1 /set radix
sas (1
ss3pqq
jjmmpp eerrrr
ddac ops
jm lsv

pb, lac pc /entry from user program
dac wrd
ral 6s
and (7
dac lc2
dzm mod
dzm lcp
lac wrd
jda pad
lac lwt
jmp ta4+5

ta4, jda pad
lio mod
law 7221
spi
law 7257 /for type-in mode
ta4+5, jda tys
lio lcp
jmp ta5+6

ta5, lac wrd
and (70000
ral 6s
dac lc2
ta5+4, lio iif
dio lcp
ta5+6, dio tsp
lio lc2
dio ts2
lac wrd
dap loc
ta9, ior (add
dac tas
jsp lct
lac mod
sza
jmp lss
jsp fet
dac lwt
bax, jea lwt /pi,opt,or lwt
jmp pn2

ta6, lac lwt
and (70000
ral 6s
dac ts2
ta6+4, lac lwt
dzm tsp
jmp ta9

dre, jsp txx /drum error
text //35/de/34//
fpr, law i 47 /flush punch and reader
arq
fpr+2, law i 51
arq
jmp lse-2

ser-3, law lse /overflow while packing sym
dap dex
dzm dff

ser, lac dff /symbol table overflow
sza
jmp dex
jsp txx
text / /35/sym ovf/34/ :/
idx dff
jmp dex

bsy, jsp txx
text / /35/busy/34//
jmp pn2

nt0, TAX /filter entries from inferior sphere
lsp ccc
jmp ent
TXA /too many
770040 /lok
dap .+2
law 21
ivk . /restart
nt0+10, law i 1
adm ccc
qit

/dispatch table - lc,uc

```
define disp low,upp
[upp-uc 44]*1000 low-uc 44
termin
```

```
define letter a,b
[b-uc 44]*1000 a
termin
```

dtb,	disp pls,pls	/space
	letter 1,quo	/1,`
	letter 2,sqo	/2,`
	letter 3,pbx	/3,\$
	letter 4,daq	/4,\
	letter 5,uni	/5,↓
	letter 6,isc	/6,←
	letter 7,pul	/7,<
	letter 10,uc8	/8,>
	letter 11,fs	/9,↑
	disp err,err	
	letter 0,arw	/0,→
	disp bar,del	//,?
	letter 34,smb	/s
	letter 35,tbl	/t
	letter 36,dec	/u
	letter 37,.vf	/v
	letter 40,www	/w
	letter 41,sxr	/x
	letter 42,.rd	/y
	letter 43,.zr	/z
	disp err,err	
	disp com,eql	/.,=
	disp err,err	
	disp err,err	
	disp tab,tab	/tab
	disp err,err	

disp .pw,ndb	/:	-
letter 23,.jb	/j	
letter 24,kil	/k	
letter 25,ttl	/l	
letter 26,m /m		
letter 27,.nw	/n	
letter 30,oad	/o	
letter 31,.pr	/p	
letter 32,q /q		
letter 33,rad	/r	
disp err,err		
disp err,err		
disp min,pls	/-,+	
disp def,bas	/),]	
disp ovb,vb /; ,		
disp lpr,bac	/[,[
disp err,err		
letter 12,a /a		
letter 13,bk	/b	
letter 14,cns	/c	
letter 15,.pn	/d	
letter 16,eee	/e	
letter 17,f /f		
letter 20,bgn	/g	
letter 21,oct	/h	
letter 22,ir	/i	
disp lc,lc /lower case		
disp dot,xec	/.,*	
disp uc,uc /upper case		
disp bs,bs /backspace		
disp err,err		
disp cr,cr /car. ret.		

lwt,	0	/last word typed
	dap pnx	
	lac lwt	
pns,	jda pi	/pi, opt, or tys, depending on S/C/T switch
px,	jmp .	
pad,	0	/print address
	dac psy	
	lac lc2	
	sza i	
	jmp .+4	
	jda opt	/other than core 0
	cli	
	tyo	
	law 7777	
	and pad	
	dac pi	
pa1,	jmp ped	/ped or poc+1, depending on R/O switch
pi,	0	/print instruction
	jda psy	
	cli	
	tyo	
	lac (i	
	and pi	
	sza i	
	jmp pid	
	cma	
	adm pi	
	law 71	
	jda tys	
	lac pi	
	sza i	
	jmp i psy	
	tyo	
pid,	lac t1	
	and (opr	
	sza i	
	jmp poc	
	sad (sft	
	jmp pnj	
	sad (law	
	jmp ped	
	sub (skp	
	rar 1s	
	sma	
	jmp poc	
ped,	jsp psy+1	
	lac (flexo +	
	jda tys	
poc,	lac pi	
	jda opt	
	jmp i psy	

```
psy, 0          /scan symbol table
dap psx
lac pi
dac t0
dac t1
and (770000
sad (770000
law 600
dap mms
and (760000
sad (sft
law 400
dap sfs
and (760000
sad (iot
law 600
dap ios
and (760000
sas (opr
sad (skp
law 600
dap psf
psa-1,
psa,
pta+1,
law i top-(0
dac t3
psr, TXXA      /examine symbol pointed to by XR
dac t4
lio i top
SXX
spi
jmp .-3
ril 1s      /t4 points to base of sym, XR points to value
spi
jmp pet      /symbol is calm
lac pi
lio i top      /symbol value
A$IP|
jmp pgx      /exact match
AMI>P
jmp pet      /sym is too big
A$IA
and (760000
sza i
sni
jmp pet
skp .
jmp pss
law i 7777  /skp or opr class
ior t0
cma
A<IP
jmp pet      /too many bits
law 7777
and t0
xct pvh
A<IP
jmp peq
jmp pet
```

pss, law 7777
and pi
dac t0 /address part of word
law 7777
and i top
mms, skp i
jmp pem /op code 77
sza i
jmp peq
sub t0
szm
jmp pet /sym > address
add (100
sfs, skp i .
add (677
sma
jmp peq
xct los /sym not close enough
jmp pet
peq, law i top-(0
sad t3
jmp pes /t3 is empty
lac i top
X→IX
lxr t3
sub i top
X→IX
spa
jmp pet
ios, skp . /this sym better than last one
jmp pes
lac pi /iot class
sub i top
and (77
sza
jmp pet
pes, lac t4
dac t2
TXXA
dac t3
pet, SXXAP|
jmp ,+3
sas (-18.
jmp psr
xct mms
jmp pmn
peu, law i top-(0
xct psf
jmp pbs
sas t3 /skp cr opr class
jmp pvh-1
lac t0
dac pi
sza
jmp psx
jmp i psy

pem, sas (70 /check for meta
jmp pet
and pi
sas (70
jmp pet
jmp peq

pvh-1, lac (flexo ↓
pvh, skp .
jda tys
lxr t3
lac i top
cma
and t0
dac t0
jdp spt
cla
jmp psa-1

pgx, lac t4
dac t2
jdp spt
jmp i psy

pbs, sad t3 /search ended
jmp poc /no acceptable symbol found
lxr t3
lac i top
cma
adm pi
jdp spt
psx, jmp .

pnj, law 1 /sft class
add pi
and pi
sza
jmp poc
lxr (-18,
jmp psa+1

pmn, law 77 /op code 77
ior pi
TAM|
jmp pm1
law 7000
and pi
sza i
jmp peu /spec. inst.
law 74
jda tys
lio pi
ril 6s
cla
rcl 2s
TAXP|
jmp mpb
rcl 2s
sza i
jmp mpe
rcl 1s
lxr i mp2-1
X>AX
ior i mp3-2
jmp mpc
rcl 3s
TAX
lac i mp1-4
mpc,
dio pi
jda tys
lio pi
TIIIX<M
ZAP
lax char ra
X+IXI>P
ior mp4
X+I>P
ior (char lx
jda tys
lio pi
ril 7s
law 17
A<IX
lac i mp5
jda tys
jmp i psy

mpe, rcl 1s /unary
rar 1s
xor (char lt
ior i mp4-1
jmp mpc

pm1, law 7254 /print negative number
jda tys
lac pi
cma
jmp poc+1

mp1, flexo z
flexo sa
flexo si
flexo sx
mp2, flexo a i
flexo x a
flexo x i
mp3, char m0
char mm
char m5
char m6
char m3
char m-
mp4, char mi
char ma
char mx
mp5, 0
flexo m
flexo p
flexo ,
flexo 7
flexo 7m
flexo 7p
flexo :7
flexo 8
flexo 8m
flexo 8p
flexo :8
flexo ;,
flexo p;
flexo m;
flexo ;

/assorted type-out routines

```
tys,      0
        dap tyx
        lxr (-3
tyl,      lac tys
        ral 6s
        dac tys
        and (77
        TAAIP]
        jmp tyc
        sas (74
        sad (72
        jmp dns
tyb,      tyo
tyc,      SXXIP
        jmp tyl
        lac lwt
tyx,      jmp .

dns,      ral 6s
        xor (skp 7400
        sad cas
        jmp tyc
        dac cas
        jmp tyb

lcc,      lio (7277
        jmp ,+2
lct,      lio (7236
        dio tys
        jnp tys+1
```

/numeric print

```
opt,      0
        dap opx
        dzm t2
opa,      lac opt
opb,      dac t3
        cli+swp
        rcl 20
        div ops
ops,      10
        sas t2
        jmp opb
        sni
        lio ops-2
        tyo
        lac t3
        dac t2
        sat opt
        jmp opa
opx,      jmp .
```

txx, dap txy
law 72
jda tys
aam
lio txy
idx txy
lac (607600
rcl 6s
sad (lai
jmp ,
sad .+2
jmp txx+3
swp
tyo
lia
jmp txy-3

/print name of symbol pointed to by t2

spt, 0
lxr t2
lac i top
and (177777
mul (1
div . 1
50
dio . 3
mul (1
div .-3
0
jdp cv1
jdp cv1
lac .-3
jdp cv1
lxr t2
idx t2
lac i top
spa
jmp spt+1
jmp i spt

/unsquoze 1 character

cv1, 0
dio t4
sza i
jmp cvx
sad (45
law 46
add (7
TAAAX
sar 3s
XMAA
sar 3s
X->AX
add i cvo
lia
tyo
cvx, lac t4
jmp i cv1

cvo,	10	-10	37
	6	-22	16

/F5 on at start → attach if possible
/F5 on at end → attached
/F6 on at end → use dcc, otherwise ivk

trc, 0 /translate core
dap gex
law 77
and trc
sas trc
jmp ge2 /drum field
and (7
sas trc
jmp err
clf 6 /computation field
rar 6s
dac pvp
ral 6s
ior (010000 /attach as core 1
lia
law 22
szf 5
ivk 15
clf 5 /can't attach
jmp gex

ge2, clf 5
rar 6s
dac pvp
and (370000
sza
sub (270000
sma
jmp err
stf 6
jmp .

gex, jmp .

dep, 0 /deposit
ZIP
fet, cli+cmi
dio dff
dap dpx
lac dep
lio tas
spi i
jmp dpx /register not open
lac tsp
sza
jmp dp2 /internal register
lac ts2
dp0, sad wh2 /entry from bpi/bpo
jmp .+4
dac wh2 /core changed
cli+cmi
dio whr
jda trc
jsp ft1
lio dff
spi i
lac dep
dac i sbu
dac dep
spi
dpx, jmp .
lac whr
lio (sbu
lxr (40
jda wri
lac dep
jmp dpx
ft1, dap ftx
law 7740
and tas
sas whr
jmp dpz
law 37
and tas
TAX
lac i sbu
jmp .
ftx, jmp .
dpz, dac whr
lio (sbu
lxr (40
jda ree
jmp ft1+1

dp2, lio dff
law 7777
and tas
sub (ac
dac tys
TAXA>P
jmp err
sub (11-ac
spa
jmp dp4 /A I X F G W * M
sub (2 /M+1,M+2, or B etc.
sma
jmp dp6 /B etc.
dp7, spi /M+1,M+2, or *
jmp dp3
law 7777 /truncate to 12 bits
and dep
jmp dp3+1
sub (nbp /B etc.
sma
jmp err
spi
jmp dp9 /fet from B
lac dep /dep in B
TAAM|
jmp dp3
and (77777
cli↓cmi
dac dep
b_=0
repeat nbp,sad bk1 b dio bk1 b b_=b 1
dp8, lac dep
ral 6s
and (7
jdp ckk
cli
lxr tys
jmp dp3
dp9, lac i ac
sma
jmp dp3+4
law charac r;
jda tys
clc
dac lwt
jmp pn2
dp3, lac dep
dp3+1, spi
lac i ac
dac i ac
dp3+4, dac dep
jmp dpx

dp4, sad (-2 /A I X F G W * M
jmp dp7 /*
sad (-4 /A I X F G W M
spi /G
jmp dp3 /fet from G, or others
jmp dp8 /dep in G, check field

/get word from buffer

pv, dap pvx
lac fa
sub pvf
TAA>P
jmp pdg
sub wc
sma
jmp pdg
lac i bbu
jmp .

pvx, law 7740 /get hunk from drum
and fa
dac pvf
pg8, law .
dac wc
sub (010000
add pvf
CAA<
cla
adm wc /to prevent wrap-around
TAX
lac pvf
lio (bbu
jda ree
jmp pv+1

/read from drum or sphere
/F6 tells which
/AC = sector address (inst part clear)
/pvp = field
/IO = core address (extend)
/XR = count (inst part clear, 0 → full)

ree, 0
dap rex
lac ree
szf i 6
jmp eer
mta /drum field
lac pvp
rex-4, A↓XA
swp
dcc
jmp dre
jmp .
rex, ior pvp /sphere
swp
mta
lac (020001
eer+4, A↓XA
ral 6s
ivk 14
jmp err
jmp rex

/write, same format

wri, 0
dap rex
lac wri
szf i 6
jmp irw
ior pvp
mta
cla
jmp rex-4
irw, ior pvp
swp
mta
lac (120001
jmp eer+4

aa,	746100	/A
	747100	/I
	742700	/X
	746600	/F
	746700	/G
	742600	/W
	747300	/*
	744400	/M
	744454	/M+1
	744454	/M+2
	746200	/B

repeat nbp-1, 746254

ab,	repeat 10,0	/A ... M
	7201	/M+1
	7202	/M+2
	0	/B

b_=.
repeat nbp-1, 7201+.-b

ac,	0	/Internal reghsters.
io,	0	
xr,	0	
fg,	0	
pc,	0	
diw,	0	
xrg,	7775	
msk,	-0	
ll,	0	
ul,	7777	
bk1,	repeat nbp,-0	
nir_=.	-ac	
repeat nbp,	0	

constants

aus,	jmp lse	/nop if auto load mode
bot,	low-top	/low end of current symbol table
loc,	add	/current location
lc2,	0	/current field
lcp,	0	/current location flop
whr,	-0	/address of stuff in buffer
wh2,	-0	
drm,	sch	/drum section currently in
ts2,	0	/current examine field
tsp,	0	/current examine flop
opc,	0	/pc saved during X
obf,	0	/bkf saved during X
obp,	0	/more breakpoint junk
bkf,	0	/on if stop was at breakpoint
pno,	1	/current process
xe2,	-0	/location of last execute
ccc,	-1	/counts enters from inferior sphere
xs2,	skp	/second `<` switch
wc,	0	/variables for pv
pvf,	0	
tsw,	repeat 6,0	
bpl,	repeat 3,0	

variables

sch=[.-1]↓37+1

/search section

sch/ law .+5-.sv
aem lsx
lac t0
lio (sza
dio whr /to make ft1 work
jmp lsx

jmp sav
jmp uns
jmp hoa
jmp obt
jmp nws+1
jmp nws
jmp eas
jmp zro
jmp vfy

sav, jdp cku /S
law 100
adm wrd
lac fa /exchange read and write fields
lio a3
dio fa
dac a3
lac a2
lio wrd
dac wrd
dio a2
dio trc
jmp uns+3

uns, jdp cku /U
lax 100
adm wrd

/fa = write addr, trc = a2 = write field
/a3 = read addr, wrd = read field
/ctt = ct = -count
/F5 is on

uns+3, jsp trc+1
szf i 5
jmp .+5
lac (010000 /drum → core
adm fa
jdp dtc
jmp lse-2

stf 5
lac wrd
jda trc
szf i 5
jmp dtd
lac (010000 /core → drum
adm a3
jdp ctd
jmp lse-2

dtd, jdp wr0 /drum → drum
nop

dta, clc
dac whr
law i bbu
add fa
sub ctt
sub (1
ior (37
SAAI
add (bbu
sub (top
sub bot
TA>
jmp dtf /will fit in core
szf i 4
law 37
and a3

lia
cma
add fa
sub (top
sub bot
szf i 4
ior (37
A+IA /largest count that will fit in core
sub ctt /and end on a sector when reading
spq
law 1 /leave at least 1 word for later
CAH
add ctt
dac ct /-number to read now
dio ctt /-number to read later
jdp dtc
lac a2
szf i 4
jda trc
law i bbu
add fa
and (7740
TAAX
dap dt7
lio t0
adm t0
CXA
adm fa
law bbu
swp
jda wri
lac fa
lxr (bbu
A\$XP|
jmp dta
lio i .
dio i 0
SXX
jmp dt7-2

dt7,
dtf,
dt8,

lio wc
lac ctt
dac ct
jep dtc
szf 4
jmp dt8
clc
dac whr
lac a2
jda trc
law i bbu /fill up rest of last sector
add fa
sad wc
jmp dt9
aed t0
dac tas
jsp ft1
aam
dac fa
idx fa

jmp dt8
dt9, lxr wc
lio (bbu
lac t0
jda wri
jmp lse-2

/core to drum or sphere transfer
/a3 = core addr
/fa = drum/sphere addr
/a3 = field
/ct = -count
/uses first 40 words of bbu

ctd, 0
jdp wr0
jmp ct9
ct6, law bbu+40
sad fa
jmp ct7
aam
lac a3
aam
dac fa
law 1
add a3
dap a3
law 1
add fa
dap fa
isp ct
jmp ct6
ct7, lio (bbu
lac t0
lxr (40
jda wri
lac ct
sza i
jmp i ctd /done
law 40
adm t0
ct9, law 37
ior ct
CAAP| /check for full sectors
jmp cd5
dac t1
and (7777
TAX
lio a3
lac t1
jda wri
lac t1
add t0
dap t0
lac t1
add a3
dap a3
lac t1
adm ct
sza i
jmp i ctd
cd5, lio (bbu
lac t0
lxr (40
jda ree
lxr (bbu

aam
lac a3
dac i 0
law 1
add a3
dap a3
SXX
isp ct
jmp , -10
lxr (40
lio (bbu
lac t0
jda wri
jmp i ctd

/drum or sphere to core transfer
/fa = core addr (updated)
/a3 = drum/sphere addr (updated)
/wrd = field
/ct = -count (destroyed)
/uses ft1, so beware of whr
/if F4 on, transmit zero

dtc, 0
clf 5
szf 4
jmp dtz
lac wrd
jda trc
us1, law 37
and a3
sza i
jmp us2 /on boundary
us1+4, lac a3
dac tas
jsp ft1
aam
dac fa
law 1
add fa
dap fa
law 1
add a3
dap a3
isp ct
jmp us1
jmp i dtc
us2, law 37 /check for full sectors
ior ct
CAAP|
jmp us1+4 /no full sectors
dac t1
and (7777
TAX /count
lio fa /core addr
lac a3 /drum addr
jda ree
lac t1
add a3
dap a3
lac t1
add fa
dap fa
lac t1
adm ct
sza i
jmp i dtc
jmp us1+4

dtz, lxr fa
dzm i 0
SXXI
isp ct
jmp .-3
dio fa
jmp i dtc

hoa, lac 010000 /H
xct xsw
dac fa
lio fa
AMI<M
TII>
jmp err
law 100
add wrd
dac a2
law top-low-1
add bot
dac ct
add fa
spa
jmp err /won't fit
dac low
dac lwt
dac fa
law top
add bot
dac a3
jdp ctd
jsp lct
lac lwt
jda opt
jmp lse-2

obt, lac 010000 /0
xct xsw
dac fa
lio fa
dio a2
AMI<M
TII>
jmp err
law 100
adm wrd
law i low-bbu-300
add a2
spa
cla
dac a3
sub a2
dac ct
add (low+1
dac fa
law i top-low
dac bot
jdp dtc /read
lac low
spa
jmp err /table extends below bottom of field?
sub a2 /-size of table
sma
jmp err
add (low+1
dac tas /origin in core
sub (bbu+300
spq
jmp err /wouldn't fit in core

/tas = base of alleged symbol table

law i low
add tas
TAX
re5, TXXP|
jmp re9
re5+2, SXX<
jmp err /bad format
lac i low-1
sma
jmp .+5
ral 1s
spa
jmp err
jmp re5+2
SXX
jmp re5
re9, law i top
dac whr /since ft1 was used
add tas
jmp ki6

```

nws,      lio (sza i  /N
nws+1,    law wsf+1  /W
          jmp .+2

eas,      law ea1      /E
          dap wsf
          dio wea
          lio chi
          spi
          jmp err
          jsp ck1
          jsp lcc

/fa = addr, trc = a2 = field
/ctt = ct = -count
/F5 is on

          jsp trc+1
          dzm lcp
          lac (010000
          dac pvf
          szf 5
          adm fa
          law wsf-1
          dap pvx
wsl,      szf i 5      /read word at address in fa
          jmp pv+1

          aam
          lac fa
wsf-1,    dac chc
wsf,      jmp .        /.+1 (W,N) or ea1 (E)
          xor wrd
          and msk
wea,
          0
          jmp wst
          lac fa      /print this word
          dap loc
          jda pad
          law 2136
          jda tys
          lac chc
          jda lwt
          jsp lcc
wst,      idx fa
          isp ct
          jmp wsl
          jmp lse-2

```

ea1, dac t2
 dzm opt
 and (760000
 sad (cal
 jmp ea2
 sub (skp
 rar 1s
 sma
 jmp wst /not addressable
ea3, lac t2
 ral 5s
 sma
 jmp ea2
 law 7777 /need to indirect
 and t2
 dac tas
 szf 5
 jmp ea5
 sub pvf
 TAX>P
 jmp ea6
 sub wc
 sma
 jmp ea6
 lac i bbu
ea8, dac t2
 lac pc
 ral 1s
 spa
 jmp ea2
 idx opt
 sas (10
 jmp ea3
 jmp wst
ea6, jsp ft1
 jmp eag
ea2, law 7777
 and t2
 jmp wsf+1
ea5, lac (010000
 adm tas
 aam
 lac tas
 jmp eag

```
zero,    lio chi      /z
        xct xsw
        spi
        jmp zr2
        lio ll
        dio fa
        lio ul
        dio ct
        lia
        jmp zr3
zr2,    spi i
        jmp .+3
        dzm fa
        law 7777
        dac ct
        lio lc2
zr3,    jsp ck2
```

/fa = addr, trc = a2 = field
/ctt = ct = -count
/F5 is on

```
        jsp trc+1
        stf 4
        szf i 5
        jmp dtd      /on drum
        lac (010000 /in core
        add fa
        TAX
        dzm i 0
        SXX
        isp ct
        jmp .-3
        jmp lse-2
```

vf_y, lio chi /V
spi
jmp err
jdp cku
dzm lcp
jsp lcc
jsp trc+1
szf i 5
jmp vf5
lac fa
lio a3
dac a3
dio fa
lac wrd
vfo, clf 5
jda trc
lac (010000
adm a3
stf 2
vf3, szf i 2
jmp vf6
aam
lac a3
vf9, dac chc
jsp pv
xor chc
and msk
sza i
jmp vf8
lac a3
szf 4
lac fa
dap loc
jda pad
law 2136
jda tys
lac chc
szf 4
jsp pv
jda lwt
jsp lct
lac chc
szf i 4
jsp pv
jda lwt
jsp lcc
vf8, law 1
add fa
dap fa
law 1
add a3
dap a3
isp ct
jmp vf3
jmp lse-3

vf5, stf 5
lac wrd
jda trc /try to attach other field
szf i 5
jmp vf7
stf 4 /to indicate reversed order
lac a2
jmp vf0

vf7, dzm ctt /can't attach either
lac a2
jda trc
xct pg8
sar ls /split buffer in half
and (7740
cma↓stf 4
dac a1 /length of top buffer
adm pg8 /make pv use lower buffer
add (bbu
dap vf2 /origin of top buffer
jmp vf3

vf6, lxr ctt /get word from top buffer
TXX<
jmp .+4
idx ctt
lac i .
jmp vf9
lac wrd /read a chunk
jda trc
law 7740
and a3
lia
sub a1
sub (010000
sma
cla /to prevent wrap-around
add (010000
AMIX
sub a3
cma
dac ctt
law . /origin of core area
A+XA
dap vf4
xct vf2
swp
jda ree
lac a2
jda trc /restore other field for pv
jmp vf6

cku, 0 /set up save or unsave
lac fa
add (400000
xct xsw
cla
dac a3 /offset
lac cku

ck1, lio ll /set up E,W,N,S,U,V
dio fa
lio ul
dio ct
lio lc2

ck2, dio a2 /set up Z
dio trc
dap xcl
cla\stf 5
dip fa /core address
dip ct
lac ct
sub fa
SAA>
jmp err
CAI
dio ct /-count
dio ctt
lac fa
add a3
and (7777
dac a3 /drum addr for S, U
xcl, jmp .

a1, 0
a2, 0
a3, 0
ct, 0
ctt, 0

/prepare for writing from bbu

```
wr0,      0
clf 5
lac a2
jda trc
law i 37
and fa
dac t0
sub fa
lio (bbu
dio fa
CAA |=
jmp i wr0
adm fa
lio (bbu
lxr (40
lac t0
jda ree
idx wr0
jmp i wr0
```

constants

[.-1]↓37+1/

szh_=.-sch

/run section

offset szh
sch/ law .+4-.ed
adm lsx
lac t0
lio chi
jmp lsx

jmp edi
jmp ff
jmp bg0
jmp xe0
jmp pra
jmp pwd

/read process state, number in I0

```
rpp, 0
law ac
mta
law 42
ivk 14
jmp i rpp
idx rpp
lio io      /put things in right order
lac xr
dac io
lac fg
dac xr
law i 7700
and pc
dac fg
dio pc
jmp i rpp
```

/write process state, number in pno

```
wpp, 0
lac ac
dac tsw
lac pc
dac tsw 1
lac io
dac tsw 2
lac xr
dac tsw 3
lac fg
dac tsw 4
lac diw
dac tsw 5
law tsw
mta
lio pno
sni
jmp .+4
law 52
ivk 14
jmp i wpp
idx wpp
jmp i wpp
```

bpo, ZIP /breakpoints out
bpi, cli+cmi /breakpoints in
 dzm t2
bpi+2, dio dff /breakpoints in except for t2
 dap bp6
 law bp5
 dap dpx /exit from fet/dep
 law i nbp
 dac t3
bp2, lxr t3
 lac i bk0+2*nbp
 dac dep /saved instruction
 lac i bk1+nbp
 spa
 jmp bp4 /no breakpoint here
 dac tas
 ral 6s
 and (7
 jmp dp0 /go to fet/dep

bp5, lio dff
 X→IM
 jmp bp4
 lxr t3
 dac i bk1+2*nbp /save instruction
 lac t2
 A\$XP
 jmp bp4 /don't put in breakpoint
 TIX
 lac (bpt
 dac i sbu
 lac whr
 lio (sbu
 lxr (40
 jda wri
bp4, isp t3
 jmp bp2
bp6, jmp .

bg0, law 7 /G
and lc2
sas lc2
jmp err
jdp ckk
rar 6s
xor pc
and (070000
xor pc
dac pc
lac wrd
dap pc
clc
dac opc /save bad pc
jmp p1

```
xe0,    lio pc      /*  
dio opc  
lio bkf  
dio obf  
lio bpl  
dio obp  
lac xrg  
xct xsw  
dac fa  
law 7  
dip fa  
and lc2  
sas lc2  
jmp err  
rar 6s  
ior fa  
dac xe2      /execute address  
xor pc  
and (077777  
xor pc  
dac pc  
lac lc2  
jda trc  
law i 37  
and fa  
dac t2  
lia  
law 42  
add fa  
and (-37  
AMIAx  
dac t1  
lai  
lio (bbu  
jda ree  
law 37  
and fa  
TAX  
lac wrd  
dac i bbu  
lac (bpt  
dac i bbu+1  
dac i bbu+2  
lac t2  
lio (bbu  
lxr t1  
jda wri  
jmp p1+2
```

pra, spi /P
law 1
cma
add (400000
dac bpl+1 /proceed count
lac (070000
and pc
ral 6s
jdp ckk
lac bkf
xor pc
and (077777
sza /proceed from bpt?
jmp p1+2 /no
law i nbp
dac t2
p8, TAX
lac i bk1+nbp
sad bpl /one still there?
jmp p9 /yes
isp t3
jmp p8
jmp p1+2
p9, cli↓cmi /must interpret bpt
jsp bpi+2
law 4000
ior fg
dac fg
jmp p2 /turn on ESI
p1, clc /start user
dac xe2
p1+2, clf 7 /disable call button
jsp bpi
law i 4000
and fg
dac fg
lac (400000
dac bpl+1 /clear proceed count
p2, jsp lcc
jdp wpp
jmp err
lio (bpl
law 72
ivk 14 /write bpl stuff
law 12
ivk 14 /permit processing
jmp nt0+10

ent, iam
eem
TXA
dap nt4 /entered process capability
dio t1
law 2
ivk 14 /suppress processing
clc
dac whr
lio (bpl
law 62
ivk 14 /read breakpoint status
jsp bpo
law 17
and t1
dac t1
sad (14
jmp cql /hit call
law 61
nt4, ivk . /read process number
hlt
lia
law 21
xct nt4 /restart it
dio pno
jdp rpp
nop
lxr t1
xct i dqq
ior (740000
lia
jmp ii+1

dqq, law 1010 /0 - illegal instruction
law 0404 /1 - lock fault
law 0606 /2 - stray ESI trap
law 2121 /3 - fcn busy
jmp 1bp /4 - breakpoint
law 1111 /5 - halt
law 0707 /6 - memory protect
jmp cl /7 - iot 2377
law 0506 /10 - mta 4
law 0605 /11 - mta 5
jmp aut /12 - mta 6 (avtomatic ac->G,K,2T,1U)
jmp cl /13 - mta 7 (dsm)

cql, lio pno
jdp rpp /try to get old proc back
skp i
jmp cll
lio (1
dio pno /use process 1
jdp rpp
skp i
jmp cll
clc
dac bkf
dac bpl
jmp npp /no proc

1bp, lac xe2 /breakpoint
TAAM| /* in progress?
jmp 2b /no
xor pc
and (070000
sza
jmp 2b /returned to wrong core
lac pc
ior (400000
sub xe2
and (7777
sad (1
jmp cl /returned,no skip
sas (2
jmp 2b /incorrect return
jsplcc /skipped
jmp cl

2b, lac bpl
b_=0
repeat nbp-1,sad bk1 b jmp 3b b_=b 1
sas bk1 b
jmp ii /bpt not assigned

3b, law 7255
dac lwt
lio pc
dio bkf
jmp ob

cli, lio (7255 /call button
jmp . 2

ii, lio (741010 /illegal instruction
clc
dac bkf
dac bpl

3bp, dio lwt
lac xe2
TAAM|
jmp pb /* not in progress
xor pc
and (077777
sza
jmp pb
law 7473 /this was an *
jda tys
lac lwt
jda tys
jsp lct
lac xe2
ior (add
dac tas
rat 6s
and (7
dac ts2
dzm tsp
jsp fet
jda lwt

cl, lac opc /dsm
xor pc
and (077777
xor pc
cl+4, dac pc /aut comes here
lio obf
dio bkf
lio obp
dio bpl
clc
dac opc
dac obf
dac obp
xct aus /nop if auto load mode
law i top-low
dac bot /K
law 2
dac wrd
jmp .tb /2T next

/automatic G addr setup, K, 2T, 1U
aut, lac (nop
dac aus
dzm lc2
lac ac
jmp cl+4

edi, cla+stf 5 /E
jea trc
szf i 5
jmp err /not in core?
cla
mta
lio (610000
lac (010000
mta 104 /read drum
jmp dre
law 2
dac pc
ide,
clc
dac opc
b_=0 repeat nbp,dac bk1+b b_=b+1
dac msk
dzm ll
law 7777
dac ul
dzm fg
jmp p1

fil, law 16 /start file sys
mta
lac (204272
ivk 14 /dismiss old tape
lac t1
ral 6s
ior (16
mta
lac (306272
ivk 14 /assign new tape
jmp bsy
dzm ac
dzm io
dzm pc
cla
jda trc
lac (ivk 16
dac bbu
dzm bbu+1
cla
lio (bbu
lxr (40
jda wri
jnp ide

ff, lac l1f /F with argument
sza
jnp err
lio let
lxr (syl-tpp
spi i
lxr (cnc-top
lac wrd
xct sgn
dac t1
spi
jmp ff2 /number
jsp lcc /symbol
lac t1
lio fa
xct xsw
cli
arq
jmp ff3
dac t2
jsp lcc
lac t2
ff3:, sad t1
jmp lse-2
ral 6s
jda opt
jmp lse-2

ff2, and (17
sad t1
jmp fil /start file sys
jsp lcc
lac t1
sad (-2
jmp m2f
SAP|
jmp m1f
and (777
lio fa
mta 1
lio fa2
ral 9s
jor (272
ivk 14
jmp .+4
dac t2
jsp lcc
lac t2
sad fa
xct xsw
jda opt
jmp lse-2

m1f, law 202 /create process
ivk 14
jmp err /can't
jdp wpp /write out old stuff
nop
law 222
ivk 14
dac pno
lrx (-6
dzm i ac+6 /clear live registers
SXXP
jmp .-2
jmp lse

m2f, lio pno /delete current process
law 212
ivk 14
jmp err
lio pno
jdp rpp
skp i
jmp lse
lio (1
dio pno
jdp rpp /try process 1
skp i
jmp lse
dzm pno
jsp txx
text //35/no proc/34//:
jmp lse

pwd, lac pno /:
spi
jmp eql
jdp wpp
nop
lio wrd
jdp rpp
jmp err
lio wrd
dio pno
jmp lse

constants

[.-1]+37+1/

szn_=.-sch

/pdv section

offset szh+szn
sch/ law .+4-.tp
adm lsx
lac t0
lio chi
jmp lsx

jmp tp0
jmp tb1
jmp jbk
jmp pun
jmp rd

jbk, spi /J
jmp err
law charac rp
arq
jmp bsy
law i 40
jdp fee
lac wrd
ior (jnp
jdp pbw
law i 520
jdp fee
jmp fpr

pbw, 0
lia
repeat 3,ppb rcl 6s
adm t2
jmp i pbw

fee, 0
cli
ppa
SAAP
jmp .-2
jmp i fee

ar, 0 /assign reader and start reading
law charac rr
arq
jmp bsy
jsp lct
skp i
rpb /flush loader
rpb
spi i
jmp .-3
jdp rbk
law flexo ok
jda tys
jsp lct
jmp i ar

rbk, 0 /read a block
dzm let
dzm chi
rpb
dio t2
dio cnc
TIA<M
jmp .+5
and (077777 /start block
szf 4
dac pc
jmp fpr+2
rpb
dio ch
lai
sub t2
spq
jmp ere
law i 1
add ch
xor t2
and (777700
sza
jmp ere
ZX
rb0, rpb
dio i rbf
lai
adm chi
SXXI
idx t2
sas ch
jmp rb0
dio iif
add chi
add cnc
rpb
A\$IP
jmp cse
cla
dip cnc
dip ch
jmp i rbk

cse, jsplxx /checksum error
text //35/cksm/34/ ://
tyi
lai
sas (charac rc
jmp .+6
law 51 /read block again
arq
jmp err
jsplct
jmp rbk+1
sas (charac rd
jmp fpr+2
jsplct /accept block as read
jmp rb5

gwd, 0
lac let
sas l1f
jmp .+3
jdp rbk
jmp gwd+1
TAX
idx let
lac i rbf
jmp i gwd

tb1, lac wrd /T
dzm dff
sas (1
jmp pot
jep ar /1T
tb3, law i 3
add bot
dac sym
jdp gwd
dac wc
jdp gwd
dac pvf
jdp gwd
dac df1
lac pvf
and (177777
lio wc
A↓IP|
jmp fpr+0 /finished
txr sym
dac i top+1
tai
end (177777
dac i top
A\$II
sza i
idx sym /two word symbol
lac (400000
adm i top
sni
jsp de+1
jmp tb3

pot, sas (2 /2T
jmp err
law 240
dap pg8 /restrict word count for pv
law 6
dac fa
clavstf 1 /enable call button
jda trc
jsp pv /get number of symbols
TAA|=
jmp gfv
CAA<
jmp err
dac tas
law 11
dac fa
jsp pv
and (1777
dac t3 /end of table
law 7
dac fa
jsp pv
and (1777
dac fa /origin of table
sub t3
mul (1
div tas
hlt
sni i
jmp err
dac t4
sub (1
spq
jmp err
sub (10
sma
jmp err

```
gfd:, lac bot
      sub t4
      dac t2
      dac sym
      law i 1
      add bot
      dac t1
      stf 4      /goes off after first word
      stf 2      /goes off after zero seen
      dam syl
gff:, jsp pv
      lia
      and (177777
      A$IP
      dio syl      /symbol is calm
      szf i 4
      ior (400000
      lxr t1
      dac i top-1
      sni↓szf 2 i
      jmp .+4
      clf 2
      lac t1
      dac sym      /truncate symbol
      clf 4
      idx fa
      law i 1
      adm t1
      sas t2
      jmp gff
      jsp pv
      dac df1
      lac sym
      sub bot
      SA<
      jmp .+4      /no name
      lio syl
      sni          /don't define if calm
      jsp de+2
      idx fa
      isp tas
      jmp gfd
      xct aus
      cla          /auto load mode
      dap xsw
      law 1
      dac wrd
      jmp .un      /1U next
```

pun, xct xsw /D
spi
jmp pd2 /a<bD or D
lio ll /aD
dio fa
lio ul
dio t4
and (7
jmp pd3
pd2:, spi i
jmp .+3
dzm fa
law 7777
dac t4
law 7
and trc
sas trc
cla
pd3:, rar 6s
dac chi /core for block origins
cla
dip fa
dip t4
idx t4
sub fa
spq
jnp err
lac lc2
jda trc

pb5, law 77 /punch one block
ior fa
SAA
dac t0
sub t4
CAA<
cla
aem t0 /end addr
pb4, skp 600 /or skp for read in mode
jmp pb1
dzm t2
lac fa
add chi
jdp pbw
lac t0
add chi
jdp pbw
pb1:, lac fa
ior (dio
xct pb4
jdp pbw
jsr pv
jdp pbw
idx fa
sas t0
jmp pb1
lac t2
xct pb4
skp i
jdp pbw /punch checksum
law i 10
jdp fee
lac t0
sas t4
jnp pb5
jmp pn2

rd, stf 4 /Y
lac wrd
xct xsw
spi
lac lc2
jda trc
lac ll
lio ul
xct xsw
jmp .+3
lac fa
lio wrd
dac syl /lower limit
dio sym /upper limit
cla
dip syl
dip sym
idx sym
sub syl
spq
jmp err
jdp ar
vy4, lac syl /process one block
sub cnc
spa
cla
dac let
adm cnc
lac sym
sub ch
sma
cla
adm iif
sub let
spq
jmp nb

```
law 100      /process block for Y
xor iif
ior let
sza
jmp yn2+4
lac cnc      /full block
lio (rbf
yn2:, lxr (100
jda wri
nb:, jdp rbk /read next block
jmp vy4

yn2+4:, law 7700
and cnc
dac ch
lio (bbu
lxr (100
jda ree
law bbu
sub ch
adm cnc
lxr let
lac i rbf
aam
dac cnc
idx cnc
SXXA
sas iif
jmp .-6
lac ch
lio (bbu
jmp yn2
```

tp0, law charac rp /L
arq
jnp bsy
law i 30
jdp fee
jsp lcc

tp1:, tyi
lai
sad (77
jmp pir
sad (75
jmp p22
sas (36
jmp tt1

pri:, law 0 /to punch in read in mode
jmp pi1

pir:, law i 40
jdp fee
lxr (-24

pi3::, lac (dio 7776
X+AA
jdp pbw
lac i lod+24
jdp pbw
SXXP
jmp pi3
jmp p22+2

p22:, law i 30
jep fee
lac (jmp 7752
jdp pbw
law 600 /to punch data blocks

pi1:, dap pb4
law i 30
jdp fee
jmp lse

lod:, eem
cli
770037 /lcr
rpb
dio 7776
TIX>P
jmp i 7776
rpb
dio 7777
rpb
dio i 7776
X+IX
idx 7776
sas 7777
jmp 7763
X+AA
rpb
A\$IP
hlt
jmp 7755

tt1, dac ch
sal 2s
adm ch
law i 5
dac t1
tt4:, idx ch
mul tt5
TAX
rcl 2s
lio i tt0
TAX
xct i tt8
law 77
A-II
ppa
isp t1
jmp tt4
cli
ppa
jmp tp1

/title punch table

tt0:, 0	0	/space,1,2	
427740	006251	515156	
224145	453214	/3,4,5	
121177	102745	454531	
364545	453001	/6,7,8	
017105	033245	454532	
065151	513600	/9	
tt8:, nop	ril 6s	rir 6s	
nop	0		
0	0	0	
000000	000036	/0,/	
414141	364020	140201	
224545	453001	/s,t,u	
017701	013740	404037	
073060	300737	/v,w,x	
601460	374122	142241	
010274	020161	/y,z	
514145	430000	000000	
141414	141400	/=	
0	0	0	
0	0		
000000	001010	741010	/:
204040	403777	/j,k,l	
101422	417740	404040	
770214	027777	/m,n,o	
021420	773641	414136	
771111	110636	/p,q,r	
415121	567711	113146	
tt5:, 125252	0		
000000	001010	101010	/-
004132	140000	/), ,(
007700	000014	224100	
000000	000076	/a,b	
111111	767745	454532	
364141	412277	/c,d,e	
414141	367745	454141	
770505	010136	/f,g,h	
415151	307710	101077	
004177	410000	/i,low,.	
000103	000000	606000	
000301	000000	/up	

constants

rbf, .+100/

[-1]↓37+1/

szv_=.sch

```
repeat if2,[printo szh      printc 77
printo szn      printc 77
printo szv      printc 77]
```

bbu=sch+szh

```
repeat ifm szh-szn,bbu=sch+szn
repeat ifm bbu-sch-szv,bbu=sch+szv
```

```
define this a
    squoze a    a
terminate
```

```
offset 0
7400/
low,      this i
this and
this ior
this xor
this xct
this lxr
this jdp
this cal
this jda
this lac
this lio
this dac
this dap
this dip
this dio
this dzm
this adm
this add
this sub
this idx
this isp
this sad
this sas
this mul
this div
this jmp
this jsp
this clo
this skp
this szs
this szf
this sqp
this szm
this szo
this spi
this sni
this sma
this sza
this spa
```

this sft
this ral
this ril
this rcl
this sal
this sil
this scl
this rar
this rir
this rcr
this sar
this sir
this scr
this law
this iot
this tyi
this ckn
this rrb
this cks
this dsc
this asc
this cac
this lsm
this esm
this cbs
this dia
this dcc
this dra
this rbt
this arq
this wat
this sdl
this lei
this lea
this rer
this rpa
this rpb
this tyo
this ppa
this ppb
this dpy
this ivk

this nop
this opr
this clf
this stf
this lat
this swp
this cmi
this clc
this cli
this cla
this cma
this lia
this lai
this hlt
this frk
this qit
this bpt
this lem
this eem
this rpf
this lpf
this nam
this bam
this iam
this dam
this aam
this mta
this dsm
this 9s
this 8s
this 7s
this 6s
this 5s
this 4s
this 3s
this 2s
this 1s

top,

start lse-2